Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets

Julie Digne, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun, Fernando de Goes

Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets

Julie Digne, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun, Fernando de Goes

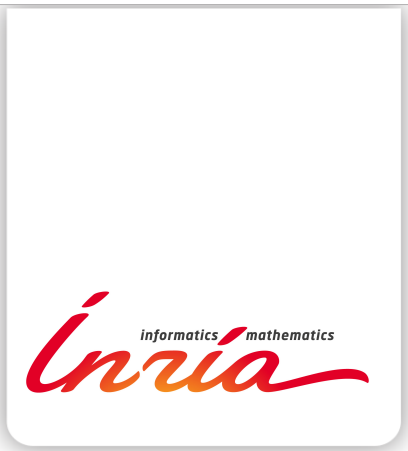
RESEARCH REPORT

N° 7991

2012 June, 11th

Project-Team GEOMETRICA

ISSN 0249-6399 ISRN INRIA/RR--7991--FR+ENG



Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets

Julie Digne*, David Cohen-Steiner*, Pierre Alliez*, Mathieu Desbrun†, Fernando de Goes†

Project-Team GEOMETRICA

Research Report n° 7991 — 2012 June, 11th — 23 pages

Abstract: We propose a robust, feature-preserving surface reconstruction algorithm which turns a point set with noise and outliers into a low triangle-count simplicial complex. Our approach starts with a simplicial complex filtered from a 3D Delaunay triangulation of the input points. This initial approximation is iteratively simplified based on the optimal cost to transport the point set to the simplicial complex, both seen as measures (or mass distributions). Our optimal transport formulation allows the recovery of sharp features even in the presence of a large amount of outliers and/or noise in the input set.

Key-words: Optimal transportation, Wasserstein distance, Linear programming, Surface reconstruction, Shape simplification, Feature recovery.

* GEOMETRICA - INRIA Sophia Antipolis
† California Institute of Technology

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Reconstruction et simplification de surfaces à partir de nuages de points imparfaits

Résumé : Nous proposons une méthode robuste de reconstruction de surface qui préserve les bords et les arêtes vives. Cette méthode part d'un nuage de points bruités et contenant des points aberrants pour reconstruire un complexe simplicial parcimonieux. Notre approche débute par la construction d'un complexe simplicial par filtrage d'une triangulation de Delaunay des points initiaux. Cette approximation initiale est ensuite itérativement simplifiée en se basant sur le coût de transport entre le nuage de points et le complexe simplicial, ceux-ci étant vus comme des distributions de masse. Cette formulation basée sur le transport optimal entre les deux distributions permet de retrouver les arêtes vives même en présence de nombreux points aberrants ou de bruit dans le nuage de points initial.

Mots-clés : Transport optimal, Distance de Wasserstein, Programmation linéaire, Reconstruction de surface, Simplification de forme, Recouvrement d'arête et de bord

1 Introduction

Surface reconstruction is a multi-faceted challenge which precise problem statement depends on the nature and defects of the input data, on the properties of the inferred surface (smooth vs piecewise smooth, with or without boundaries) and on the level of details sought after. Despite a number of major contributions on this topic over the past decade [3, 19], achieving both feature preservation and robustness to measurement noise and outliers remains a scientific challenge—and a pressing requirement for many reverse engineering and geometric modeling applications. Furthermore, low polygon-count reconstructions has only received limited attention despite the increase of point density in 3D scanning technology and the need for efficient further geometry processing.

In this paper we contribute a reconstruction method that simplifies an initial (possibly non-manifold) triangulation of the input point set based on an error metric that quantifies the distance between the current simplicial complex and the input points through optimal transport of measures. Our reconstruction approach inherits the qualities of the optimal transport derived metric, i.e., resilience to noise, outliers, and imperfect sampling, as well as sensitivity to boundaries and sharp features (creases, corners, tips, etc). We demonstrate these distinguishing properties on a series of examples. The proposed metric can also be used as a post-processing tool for recovering sharp features and boundaries from the output of a smooth reconstruction method.

2 Previous Work

We first discuss existing surface reconstruction methods, restricting our review to approaches that are both robust to noise and outliers and also feature preserving. We then point out recent, relevant work on geometry processing based on optimal transport.

2.1 Surface Reconstruction

A common approach to robust surface reconstruction from defect-laden point sets involves denoising and filtering out outliers, and often requires an interactive adjustment of parameters. Automatic methods such as spectral methods [20, 40, 2] and graph cut approaches [16, 21], on the other hand, are shown to be extremely robust, but are better suited to the reconstruction of smooth, closed surfaces. More recently, Cohen-Or and co-authors have proposed an excellent series of contributions based on robust norms and sparse recovery [24, 17, 4]. An interpolating, yet noise robust approach was alternatively proposed by Digne et al. [9] through the construction of a scale space.

Feature preserving methods are typically based on an implicit representation that approximates or interpolates the input points. In [10], for instance, sharp features are captured through locally adapted anisotropic basis functions. Adamson and Alexa [1] proposed instead an anisotropic moving least squares (MLS) method using ellipsoidal mapping functions based on principal curvatures. More recently, Oztireli et al. [28] used a kernel regression technique to extend the MLS surface reconstruction, allowing for much sharper features. However, none of these techniques returns truly sharp features: reconstructions are always semi-sharp, with various degrees of smoothness depending on the approach and the sampling density. Moreover, sharpness is always determined locally, often leading to fragmented creases, so the reconstruction quality degrades quickly if defects and outliers are added.

Another way to detect local sharpness in a point set consists in performing a local clustering of estimated normals [27]: if this process reveals more than one cluster of normals, then the algorithm fits as many quadrics as the number of clusters. Improved robustness is achieved in [12]

by segmenting neighborhoods through region growing. Lipman et al. [23], instead, proposed a systematic enrichment of the MLS projection framework with sharp edges driven by the local error of the MLS approximation. Again, the locality of the feature detection can generate fragmented sharp edges— much like general feature detection approaches (e.g., [15, 29]).

To reduce fragmentation a different thread of work aims at extracting long sharp features. Pauly et al. [30], for instance, used a multi-scale approach to detect feature points, and construct a minimum-spanning tree to recover the most likely feature graph. Daniels et al. [8] used a robust projection operator onto sharp creases, and grew a set of polylines through projected points. Jenke et al. [18] extracted the feature lines by robustly fitting local surface patches and by computing the intersection of close patches with dissimilar normals.

[38] detects features on the point set and uses a Delaunay refinement to get a feature preserving model. This method can also be used on a model built through a noise-robust method such as Poisson [19] for enhancing sharp features of the model. We will show that the metric introduced for reconstruction can also be used in this setting.

Distinctive Approach. In this paper, we adopt a very different methodology: we reconstruct a shape through an iterative, feature-preserving simplification of a simplicial complex constructed from the input point set. To achieve unmatched noise and outlier robustness, the error metric driving the simplification is derived in terms of optimal transport between the input point set and the reconstructed mesh. Next we provide a brief review of optimal transport, and mention its applications to various problems in computer graphics and computer vision.

2.2 Optimal Transport

The problem of transporting a measure onto another one as a way to quantify their similarity has a rich scientific history [39]. For two measures μ and ν defined over $\mathbb{R}^3$ and of equal total mass (i.e., their integrals are the same), the optimal transport from μ to ν consists in finding a transport plan π that realizes the infimum:

$$\inf \left\{ \int_{\Omega^2} \|x-y\|^2 \mathrm{d}\pi(x,y) \middle| \, \pi \in \Pi(\mu,\nu) \right\},$$

where $\Pi(\mu,\nu)$ is the set of all possible transport plans between μ and ν, and L_2 norm is used as the transport cost.

The formulation above is particularly well suited to compare 1D measures such as histograms over the real line or on the circle [33], and it has been used for transferring color and contrast between images [36, 32]. For applications in higher dimensions such as 2D and 3D shape retrieval [37, 34] and segmentation [31], the optimal transport formulation is notoriously less tractable. Attempts have been made to design computationally easier surrogates, such as the *sliced Wasserstein* approach [35], which consists in projecting the transport problem on a series of 1D problems.

An alternative and more general strategy to solve the optimal transport problem is based on linear programming. Such numerical tool has been used in applications such as surface comparison [25] and displacement interpolation [5].

Distinctive Approach. Our reconstruction method also relies on a linear programming scheme. However, our approach introduces a key distinctive property: our target measure ν is *not* given, but instead, solved for. More specifically, we search for the simplicial complex of a user-specified size that minimizes the cost of transporting the input pointwise measure (i.e., the initial point set) to the complex simplices.

The closest work to ours was proposed by de Goes et al. [14]. Their algorithm reconstructs and simplifies 2D shapes from point sets based also on optimal transport. Nonetheless, their

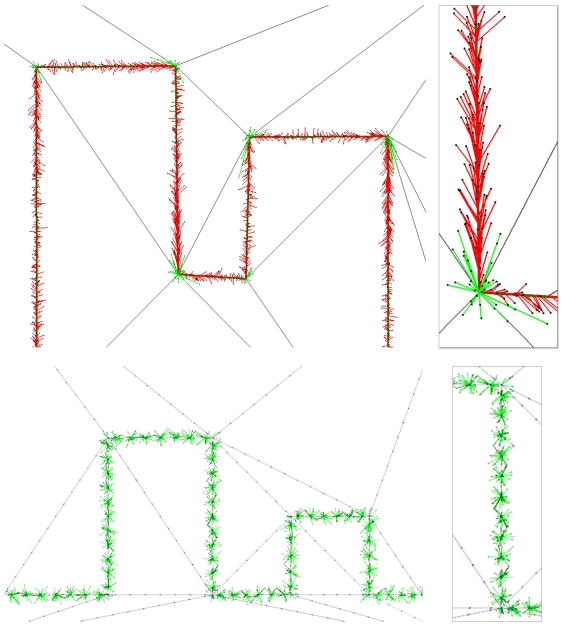


Figure 1: Transport plans. Top: Binary transport plan [14]. We depict the input point set and simplified triangulation. Red and green line segments depict the transport plan between the input point set and the uniform measure on respectively the vertices and edges of the triangulation. Each input point is simply transported either to its closest edge or to the end vertices of that edge. For each edge the transport plan favors a transport to its end vertices instead of to the whole edge when the corresponding transport cost is lower. The closeup image reveals that near the corner vertex transporting the nearby points to the vertex instead of the nearby edges creates a lower density near the end of that edge, and hence induces a tangential transport cost (red line segments pointing down) and consequently a higher total transport cost. For complex features such as concave corners on surfaces such behavior leads to incorrect reconstructions. Bottom: our transport plan is globally normal to the edges.

approach differs from ours in several aspects:

1. their optimal transport involves only points and edges and therefore can be computed in closed form. To our knowledge, no such closed form exists when transporting points to the facets of a simplicial complex. Therefore we use a discretized formulation of the optimal transport problem.

2. the authors proposed to approximate the optimal transport plan by assigning each input point to its closest edge in the triangulation. Such simplistic scheme can lead to sub-optimal transport plan and cost as illustrated in Figure 1(top); even more so in 3D. Our discretized formulation, combined with a linear programming solver, provides better approximations of both the optimal plan and the optimal cost.

3. their method requires a valid embedding of a 2D triangulation, which they achieved through a recursive edge flip procedure. Such edge flip procedure can not, however, be generalized to 3D triangulations. Alternatively, our method removes the embedding requirement by

only employing a (possibly non-manifold) simplicial complex, initially chosen as a subset of a 3D Delaunay triangulation.

2.3 Overview

Motivated by the concept of reconstruction introduced in 2D by de Goes et al. [14], our 3D method consists in a fine-to-coarse algorithm which reconstructs a surface from a point set through greedy simplification of a 3D simplicial complex. We initialize the complex with a (possibly non-manifold) subset of the 3D Delaunay triangulation of input points, then perform repeated decimations based on half-edge collapse operations. The error metric guiding our simplification is derived from the optimal cost to transport the input point set (seen as Dirac measures) to a constant-per-facet measure defined over the simplicial complex. At each iteration, we collapse the half-edge which minimizes the increase of the total transport cost between the input points and the reconstructed triangulation. Similarly to the formulation presented in [14], our optimal transport driven metric brings rare, but desirable properties, such as resilience to noise and outliers, and preservation of boundaries and sharp features.

In the remainder of this paper, we discuss the details of our optimal transport based metric (Sec. 3) and then describe our reconstruction algorithm step by step (Sec. 4). Our method is summarized in Algorithm 1 and its main stages are illustrated in Figure 2.

Algorithm 1: Algorithm Overview.

Input : Point set $\mathcal{S}$, user-specified number of vertices.
Output: Simplicial complex $\mathcal{C}$.
Construct 3D Delaunay Triangulation $\mathcal{T}$ from $\mathcal{S}$;
Compute transport cost from $\mathcal{S}$ to the facets of $\mathcal{T}$;
Construct simplicial complex $\mathcal{C}$ from all facets of $\mathcal{T}$ with non-zero measure;
Decimate $\mathcal{C}$ until desired number of vertices;
Filter out facets of $\mathcal{C}$ by thresholding mass density.

3 Optimal Transport as a Linear Program

Our reconstruction method consists in finding a coarse surface simplicial complex $\mathcal{C}$ that minimizes the optimal transport cost between the input point set $S = \{p_i\}_{i=1\cdots N}$ (with associated measures $\{m_i\}$) and a piecewise uniform measure on the facets of $\mathcal{C}$. In [14], the optimal transport cost between $\mathcal{S}$ and $\mathcal{C}$ is efficiently computed based on closed form expressions for the optimal cost between points and edges. However, to our knowledge, such closed form can not be extended between points and triangles. We thus adopt a discretized formulation of the optimal transport between $\mathcal{S}$ and $\mathcal{C}$.

3.1 Discretization

We approximate the optimal transport cost between the input point set $\mathcal{S}$ and the simplicial complex $\mathcal{C}$ using quadrature points, which we call hereafter *bins*. To this end, we split vertices, edges, and facets of $\mathcal{C}$ into a set of bins $\mathcal{B}$ and evaluate the optimal cost between $\mathcal{S}$ and $\mathcal{B}$ as the sum of squared distances between the points in $\mathcal{S}$ and the centroid of the bins in $\mathcal{B}$. We discretize the simplices of $\mathcal{C}$ as follows: a vertex is considered as its own (single) bin; an edge segment is decomposed into an integer number of equally-spaced bins; and a triangular facet is tiled by

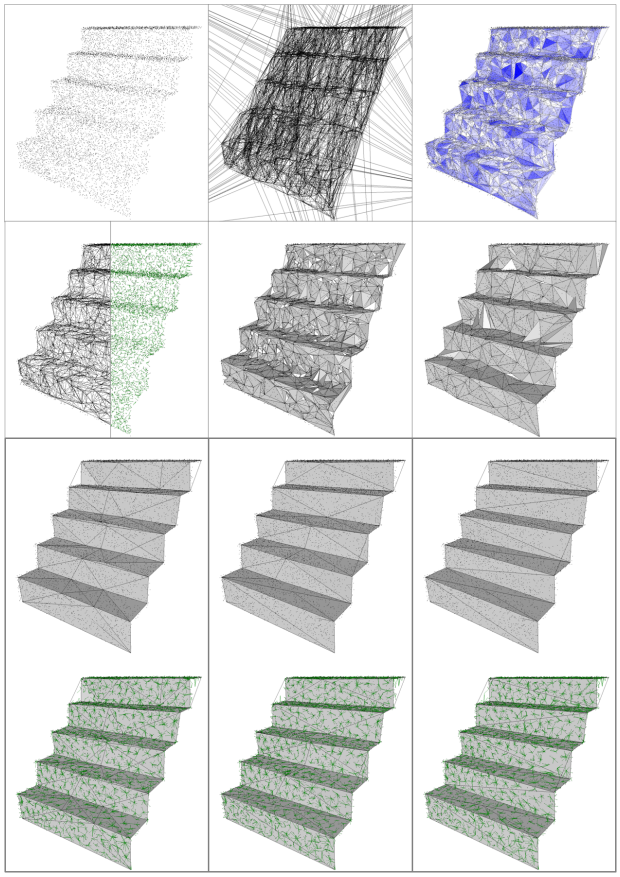


Figure 2: Steps of our algorithm: (a) Initial point set; (b) 3D Delaunay triangulation of a random subset containing 10% of the input points; (c) Initial simplicial complex constructed from facets of the 3D triangulation with non-zero measure; (d) Initial transport plan assigning point samples to bin centroid (green arrows); (e-f) decimation steps; (g-i) Reconstruction with 100, 50, and 14 vertices, respectively; (j-l) Final transport plan with 100, 50, and 14 vertices, respectively.

a 2D Centroidal Voronoi Tessellation (CVT) with bins centered at the CVT sites (Figure 3). Note that CVT meshes minimize the approximation error given by quadrature points [11] and therefore it is the optimal point-wise discretization for the simplices in $\mathcal{C}$. The number of bins per edge and facet is set based on a user-defined quadrature parameter (in our experiments, we used 200 bin per unit area). In order to accommodate the nearly uniform distribution of bins, we assign a *capacity* for each bin in $\mathcal{B}$ (i.e., the total amount of mass that a bin can receive). While we set vertex bins to unit capacity, we assign edge bins proportional to the edge length and facet bins proportional to the area of the Voronoi cells.

3.2 Linear Programming Formulation

We now present a linear programming formulation to compute the optimal transport cost between the input point set $\mathcal{S}$ and the bin set $\mathcal{B}$. In the following, we denote the simplices of $\mathcal{C}$ as

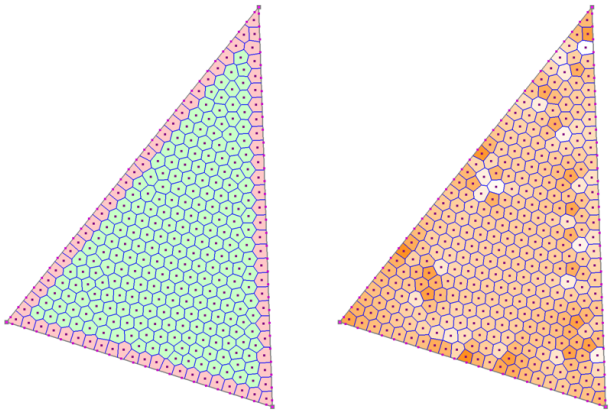


Figure 3: Bins. Left: bins in a facet. Vertices have a single bin each. Edges have equally-spaced bins. Facets have evenly placed bins constructed through a Centroidal Voronoi Tessellation. Inner Voronoi cells are depicted in green. Voronoi cells clipped by the facet boundary are depicted in pink. Bin centroids are depicted as red dots. Right: capacities of the bins are depicted using a thermal color ramp.

$\{\sigma_j\}_{j=1\cdots L}$ and the bins in $\mathcal{B}$ as $\{b_j\}_{j=1\cdots M}$, where L and M are the number of simplices and bins respectively. We also define $s(j)$ to be the index of the simplex containing the bin b_j (i.e., $b_j \in \sigma_{s(j)}$). The capacity c_j of bin b_j is defined as the ratio between the bin's area and the area of its containing simplex. Finally, we denote m_{ij} as the amount of mass transported from point $p_i \in S$ to the centroid of bin b_j (Figure 4).

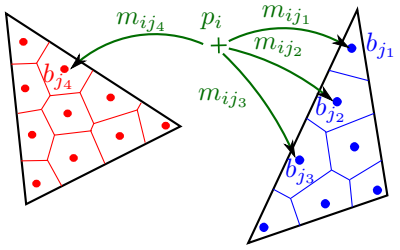


Figure 4: Transport plan for a single input sample point p_i. Variable m_{ij} models the transport of p_i to the j^{th} bin of the facet.

With these definitions, we can now formally refer to a *transport plan* between $\mathcal{S}$ and $\mathcal{B}$ as a set of $N \times M$ variables m_{ij} such that:

$$\forall ij : \; m_{ij} \geq 0, \tag{1}$$

$$\forall i : \; \sum_j m_{ij} = m_i, \tag{2}$$

$$\forall j_1, j_2 \text{ s.t. } s(j_1) = s(j_2) : \; \frac{1}{c_{j_1}} \sum_i m_{ij_1} = \frac{1}{c_{j_2}} \sum_i m_{ij_2}, \tag{3}$$

where Equation 2 ensures that the entire measure of a point gets transported onto simplices, and Equation 3 ensures a uniform measure over each simplex of $\mathcal{C}$.

An *optimal transport plan* is then defined as a transport plan π that minimizes the associated transport cost

$$cost(\pi) = \sum_{ij} m_{ij} \|p_i - b_j\|^2.$$

Finding a transport plan minimizing the transport cost results in a linear program with respect to the m_{ij}, with equality (Eq. 2 and 3) and inequality constraints (Eq. 1). Note that the number of bins, the position of bins, as well as the square distances between input points and bin centroids are all precomputed. In order to enforce the uniformity constraint (Eq. 3), we also introduce L additional variables l_i (one per simplex σ_i) indicating the target measure density of the corresponding simplex. The final formulation is thus:

$$\text{Minimize} \quad \textstyle\sum_{ij} m_{ij} \|p_i - b_j\|^2$$

w.r.t. the variables m_{ij} and l_j, and subject to:

$$\begin{cases} \forall i : \displaystyle\sum_j m_{ij} = m_i \\ \forall j : \displaystyle\sum_i m_{ij} = c_j \cdot l_{s(j)} \\ \forall i,\ j : m_{ij} \geq 0,\ l_j \geq 0 \end{cases}$$

3.3 Local Relaxation

The formulation described above is compute-intensive due to the number of variables and constraints involved. Solving this linear programming system requires instantiating a dense matrix (representing the constraints) of size $(M \times N + L) \times (M + N)$. For example, computing the optimal transport cost between a simplicial complex containing 782 simplices on which $7,300$ bins are sampled and an input point set of $2,100$ samples involves solving for a linear program of over 15 million variables and $9,000$ constraints. The linear programming solvers at our disposal do not scale to such large numbers.

In order to improve scalability we propose a stencil-based local relaxation strategy (summarized in Algorithm 2) for approximating the global optimization problem. A subset of the global solution space is explored through local solves, until a local minimum of the objective function is reached. Note that we cannot guarantee convergence of this local procedure to the global minimum, but the minima reached in practice have been sufficient in all of our tests to obtain satisfactory results.

Our procedure starts with a trivial transport plan which transports each input point to its nearest vertex of the simplicial complex $\mathcal{C}$. Since no uniformity constraints are imposed on vertices, this transport plan is valid (but obviously suboptimal). Subsequent local optimizations can only decrease the global transport cost or leave it unchanged, as the rest of the plan is kept intact and the local re-assignments have smaller or equal cost after optimization. The transport cost found by local stencil updates is thus an upper bound of the global optimal transport cost. Our experiments showed, unsurprisingly, that convergence depends on the shape of local stencils used: the larger the stencil, the faster the convergence—but with the unfortunate side effect that large stencils increase the size of the corresponding linear program. We found in practice that simply using the 1-ring of a chosen simplex is always a reliable choice. More precisely, the local stencil $\mathcal{N}$ of a facet σ includes all the facets incident to σ along with their vertices.

Algorithm 2: Solving for a local stencil.

Input : A simplicial complex $\mathcal{C}$ and point set $\mathcal{S}$, a threshold ε
Output: A locally optimal transport plan $\pi = \{m_{ij}\}$
for $p_i \in \mathcal{S}$ **do**
  Transport p_i to nearest vertex $v \in \mathcal{C}$;
$\delta = \varepsilon + 1$;
while $|\delta| > \varepsilon$ **do**
  for $\sigma_j \in \mathcal{C}$ **do**
    Build the stencil $\mathcal{N}$ of the facet σ_j;
    Collect sample points and partial measures $\{p_i, \tilde{m}_i\}$ that project onto this stencil;
    Solve the linear program for the optimal transport plan of $(p_i, \tilde{m}_i)$ on the bins of $\mathcal{N}$;
    Update transport plan π and global cost;
  $\delta = new\_cost - old\_cost$;

Armed with this scalable approximation of the transport cost, we describe next how we put it at work for surface reconstruction through simplification.

4 Reconstruction through Simplification

4.1 Initialization

We begin our reconstruction process by randomly picking a subset of the input points $\mathcal{S}$ and computing a 3D Delaunay triangulation. We then construct a simplicial complex $\mathcal{C}$ from a subset of facets of this 3D triangulation. To select this subset of facets, we perform two steps: (1) we reuse the local stencil relaxation method (Algorithm 2) to estimate a transport cost from all the input points onto the facets and vertices of the 3D triangulation; (2) we then build $\mathcal{C}$ with only the facets containing non-zero transported measure. For the step (1), we use a stencil centered at each facet and containing vertices and edges of the two tetrahedra adjacent to the facet. For an inside facet of the triangulation, for instance, this stencil contains 7 facets and 5 vertices. This stencil-based optimization is repeated in batches until the decrease in transport cost is below a user-specified threshold (set to 10^{-5} in all our tests). In practice, the global transport cost decreases rapidly, and only 10 batches are usually needed. For step (2), we convert our data structure to a simplicial complex for two main reasons: first to allow our reconstruction to have long and anisotropic simplices; and more importantly, to remove the difficult issue identified in [14] of keeping the triangulation with a valid embedding.

4.2 Decimation

From the initial simplicial complex $\mathcal{C}$, we further simplify the reconstruction through a greedy decimation based on *half-edge collapse operations*. Note, however, that a conventional decimation algorithm (e.g., [13, 22] and variants) can not be applied in our setup: the presence of outliers and noise renders the typical error metrics inadequate.

Our optimal transport framework provides a solid alternative: we pick the next half-edge to collapse as the one that increases the least the global transport cost. To this end, we simulate the collapse of a candidate half-edge e and evaluate the induced change of transport cost Δ. Since

this change cost affects mainly a neighborhood Ω_e of e, we can restrict the computation of Δ only to Ω_e. More specifically, setting Ω_e to the closure of simplices in the 1-ring of e, we first gather the set of samples p_i projecting (partially or entirely) on Ω_e (Figure 5), compute the transport cost of this set of samples to Ω_e, simulate the collapse of e, and recompute the cost of transporting the set of samples onto the resulting simplices. The change of transport cost Δ is then set as the difference of transport cost achieved before and after the simulated collapse of e. Once a half-edge is collapsed, we also update the transport plan of the edges with 1-ring intersecting the collapsed edge. Finally, to increase scalability, we employ a multiple choice approach [41] to select the next half-edge to be collapsed (as recommended in [14]), instead of maintaining an compute-intensive global priority queue. The edge selection process is summarized in Algorithm 3.

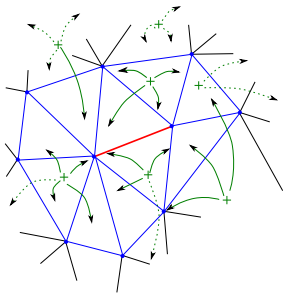


Figure 5: Local stencil of an edge. For better depiction we represent a manifold neighborhood of an edge (in red) and do not depict the bins. Simplices in the local stencil are depicted in blue. Point samples are depicted in green. We only solve for the measures transported from and to the stencil (solid green lines) and not for the measures transported outside of the stencil (dash green lines). Note that different transport plans are possible between point samples and bins, i.e., from inside to inside the stencil, inside to outside, and outside to inside.

Algorithm 3: Decimation algorithm.

Input : A simplicial complex $\mathcal{C}$, input point set $\mathcal{S}$, the number of desired vertices V
Output: A simplicial complex $\mathcal{C}_{final}$ with V vertices
for *each edge* $e \in \mathcal{C}$ **do**
  Simulate two half-edge collapse operators; Push the corresponding half-edges to a priority queue $\mathcal{P}$, sorted according to the change of transport cost Δ.
repeat
  Pop half-edge e^* out of $\mathcal{P}$;
  Collect set $\mathcal{E}$ of edges whose neighborhood intersect neighborhood of e^*;
  Collapse e^* and update transport plan on the neighborhood of e^*;
  Update $\mathcal{P}$ by recomputing the change of cost Δ for all edges in $\mathcal{E}$.
until *The simplicial complex has* V *vertices*;

4.3 Vertex Relocation

So far our method based half-edge collapses limits our reconstruction to be an interpolation of a subset of the input points. This may lead to unsatisfactory results specially in the presence of noise and outliers. We thus couple our decimation with an optimization procedure in order to relocate the position of vertices in the reconstructed simplicial complex $\mathcal{C}$. After the collapse

of a half-edge e, the remaining vertex v of e is relocated by iterating two steps: (1) for a given transport plan π, we move v toward the position that improves the optimal cost of π; (2) we update π around v. For the first step, we compute the locally optimal position of v when the transport plan π (i.e., m_{ij}) is kept fixed. To this end, we express the position of each bin (be it a vertex bin, an edge bin, or a triangle bin) in barycentric coordinates within its containing simplex. For example, finding the optimal position of vertex v of triangle $t = (v, v_1, v_2)$ amounts to minimizing:

$$\min_v \sum_i \sum_j m_{ij} \|p_i - \alpha_j v - \beta_j v_1 - \gamma_j v_2\|^2,$$

where $\alpha_j, \beta_j, \gamma_j$ are the barycentric coordinates of bin b_j with respect to vertices (v, v_1, v_2). The optimal position with respect to triangle t is thus

$$v_s^\star = \frac{\sum_i \sum_j m_{ij} \alpha_j (p_i - \beta_j v_1 - \gamma_j v_2)}{\sum_i m_{ij} \alpha_j^2}.$$

For the second step, we freeze the vertex locations and update the transport map π by resolving locally the linear program associated to the local transport (Algorithm 2). By alternating these two steps, the vertices move to their optimal position, allowing for the recovery of sharp features and surface boundaries. Figure 6 depicts a simple vertex relocation sequence in 2D.

4.4 Facet Filtering

After the decimation terminates, we return as our final reconstructed mesh the subset of facets from $\mathcal{C}$ with a non-zero measure. Since some facets may have non-zero measure due the presence of noise and outlier, we instead sort the facets based on the measure density (i.e., the ratio of facet measure and its area) and provide the user with an interactive slider to recover the most dense facets. Figure 7 shows the reconstructed surface obtained with different filtering threshold.

4.5 Experimental Results

We implemented our algorithm in C++ using CGAL's 3D Delaunay triangulation library [6] to initialize the reconstruction, and our own data structure for simplicial complexes. We used the Coin-OR Clp library [7] as our linear programming solver. Our implementation is partially parallelized to accelerate computations, exploiting the fact that all half-edge collapse simulations are independent. The initialization and update of the priority queue are, by far, the most costly operations, as each collapse involves around 120 simulations on average. When using the exhaustive priority queue on a two-core laptop, a point set containing $30,000$ points is reconstructed in around 10 hours (initial and final simplicial complexes containing respectively $3,000$ and 200 vertices). On a 8-core processor, this computation reduces to 2 hours. However, when using a multiple-choice approach with random sets of 40 collapses, the timings are three times faster on average. Memory consumption remains low; e.g., for the experiment mentioned above, the peak memory usage was around $80Mo$.

Robustness to noise. We tested our method on a point set of a staircase with an increasing amount of uniform synthetic noise (Figure 8). Even in the presence of a large noise the method tends to recover well the creases of the input shape. Only for noise magnitudes larger than 5% of the bounding box size does our method fail (for these high noise levels, spurious facets appear which cannot be discarded by a simple thresholding based on mass density). Our method can thus robustly handle inputs from any current point acquisition devices, which all have much smaller noise magnitudes.

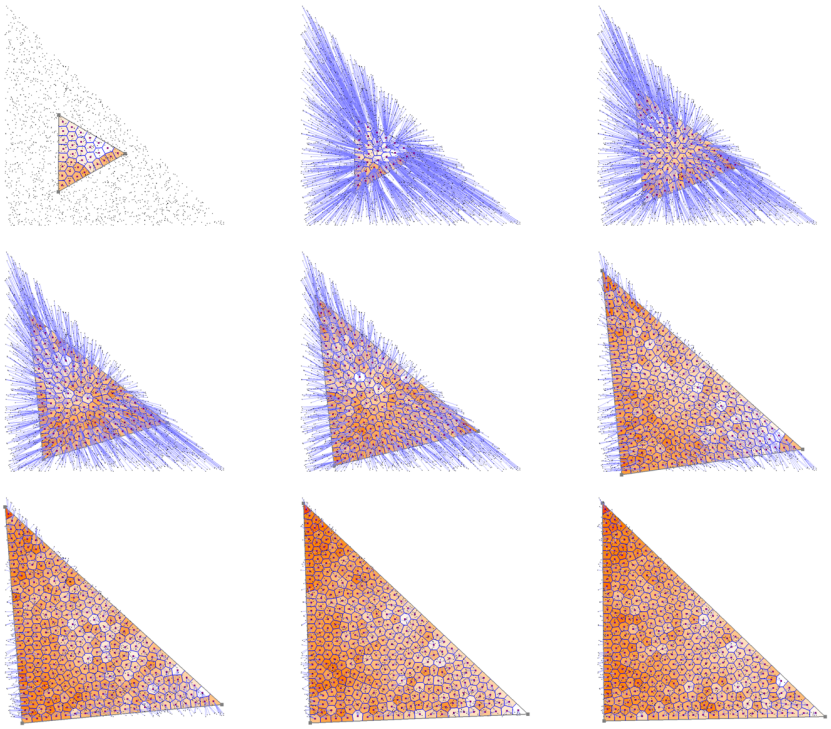


Figure 6: Vertex relocation. For visual clarity we choose a 2D example with a single triangle and only facet bins. We first depict the input point set, here uniformly sampled on a triangle, the initial simplicial complex composed of one facet, and the facet bins and their capacities. For all subsequent images we depict during vertex relocation the transport plan with blue edges connecting the source point samples and their target bin centroids.

Robustness to outliers. We also tested our method on a point set that samples a cylinder (Figure 9). Results are excellent up to 10% of outliers, but our method can fail when the amount of outliers exceeds 20%—again, current acquisition devices are usually good enough not to reach this amount of outliers.

Feature preservation. Figure 10 depicts the feature preservation property of our approach on the blade model. Our approach performs well even on thin features subtending small angles, where implicit approaches (here, the noise-robust Poisson surface reconstruction method [19]) tend to smooth out features and create spurious topological artifacts in presence of insufficient density in the input point set. On easier features subtending large angles, both approaches produce similar results when comparing our approach to the combination of a smooth implicit reconstruction followed by surface mesh decimation [22].

On a cone model (Figure 11) all features (tip, boundaries) are preserved and the simplification is very effective. Similarly, on a cylinder model (Figure 12) the boundaries are preserved and the simplification leads to anisotropic triangles with most edges aligned to minimum curvature directions as expected.

Figure 13 also illustrates boundary and sharp feature preservation. Figure 14 illustrates the behavior of our approach on two intersecting planar polygons. The algorithm behaves well down

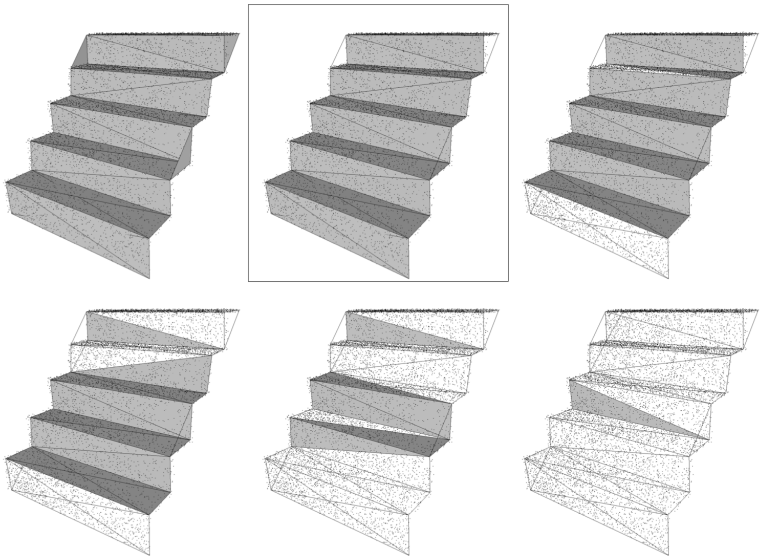


Figure 7: Facet filtering. We first depict the input point set and all facets with nonzero measure of the simplicial complex returned by our decimation scheme. We then filter out the spurious facets by thresholding facets with low measure density. The final reconstruction is highlighted in a frame.

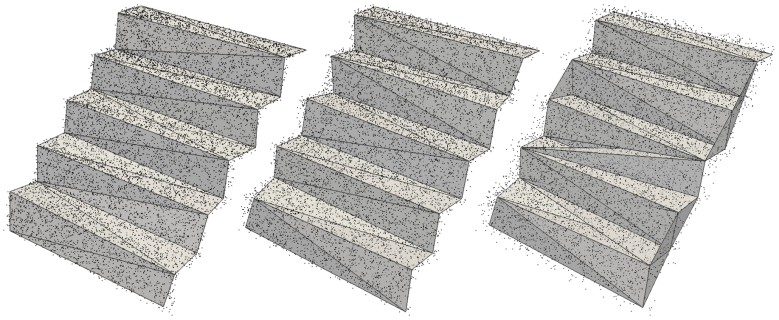


Figure 8: Robustness to noise. We increase the amount of synthetic noise from $\sigma = 1\%$ to $\sigma = 2\%$ through $\sigma = 5\%$, expressed in percentage of the longest edge length of the bounding box. The reconstruction starts failing at $\sigma = 5\%$.

to 10 vertices, and the simplicial complex maintains the initial connectivity during decimation. Going down to 8 vertices (the expected minimum number of vertices) would require a richer set of topological operators in order to disconnect the intersecting edge before pursuing decimation; we did not pursue this particular property.

Weaknesses. Given the current linear programming solvers, results of our approach come at the price of intensive computations, preventing its use on large point sets. Also, there is currently nothing in our formulation that favors 2-manifoldness, as the main data structure is a simplicial complex initialized by the facets of a 3D triangulation; this can lead, in rare occasions, to invalid embedding as well as multiple facets covering the same area (see Figure 15). The latter issue is more complex than just ensuring a 2-manifold reconstruction, as complex features may

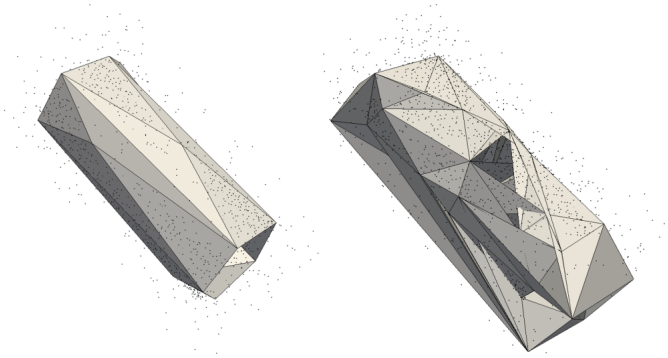


Figure 9: Robustness to outliers. The reconstruction is effective with 10% outliers (left) but fails from 20%. The outliers are added randomly within a loose bounding box (120%) of the input point set.

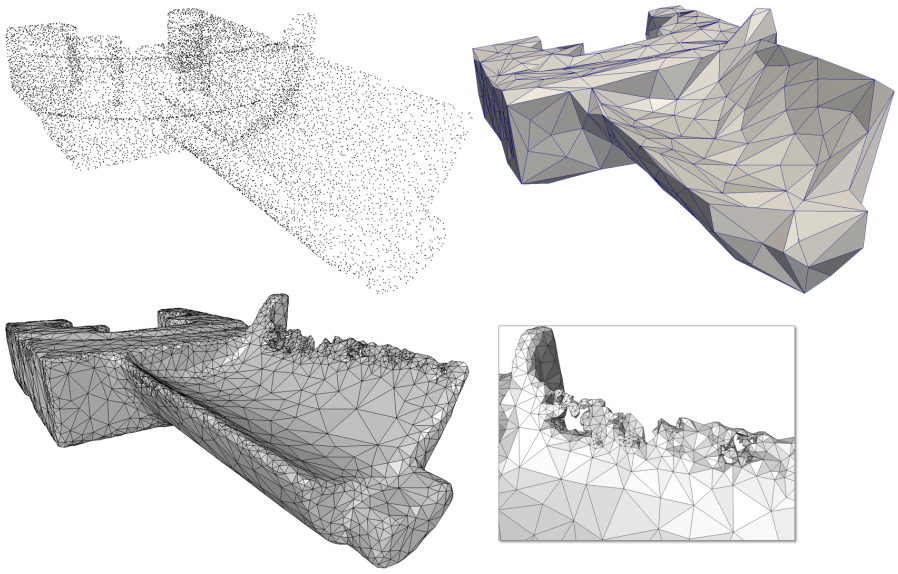


Figure 10: Reconstruction of the blade model containing $30K$ sample points. Top: our reconstruction. Bottom: the output of the Poisson reconstruction method (Delaunay refinement being used for contouring the implicit function), and closeup on a sharp crease subtending a small angle, where the implicit approach fails.

correspond to non-manifold shapes. One could define a notion of "effectiveness" per facet favoring a high transported mass and a low transport cost, but this notion would lead to a non-linear objective function and would require a richer set of topological operators such as facet deletion.

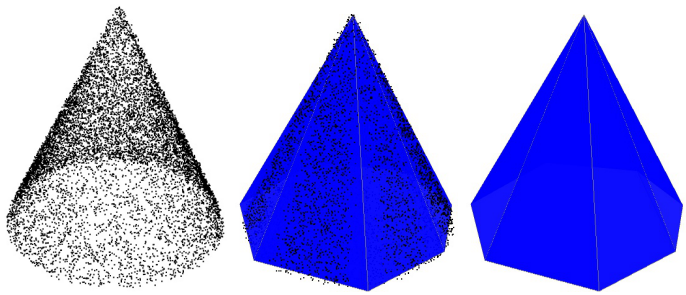


Figure 11: Reconstruction of a cone. Left: input point set Middle: input point set and final simplicial complex with (nearly uniform) facet densities shown. Right: final simplicial complex.

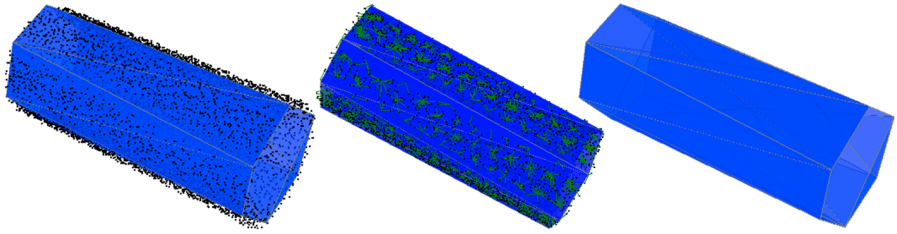


Figure 12: Reconstruction and simplification of a cylinder. Left: $10K$ noisy sample points and reconstruction with 12 vertices (facet density shown). Middle: transport plan between point samples and bin centroids. Right: simplicial complex and facet density.

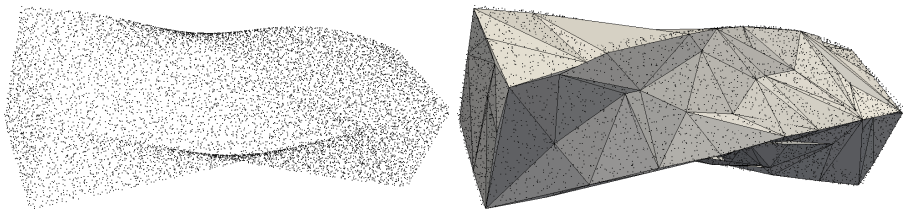


Figure 13: Reconstruction of a twisted bar. Sharp features are well preserved.

5 Feature Recovery

Another application of the proposed metric is to recover sharp features and boundaries from the output of a surface reconstruction specialized to smooth, closed surfaces (e.g., Poisson reconstruction [19]). These approaches are in general noise robust and scalable, but smoothen sharp features and fill holes. For this application we use the vertex relocation and facet filtering steps

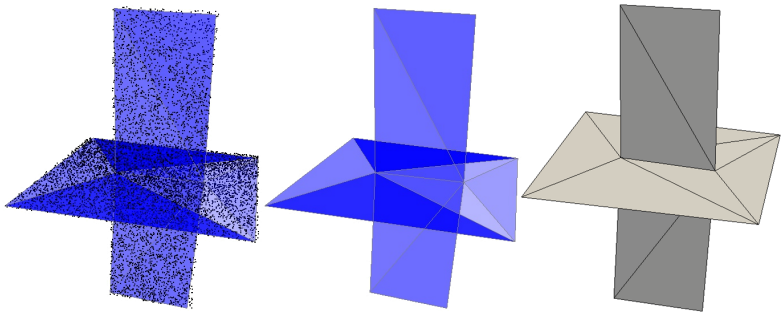


Figure 14: Reconstructing and simplification of two intersecting planar polygons until 10 vertices.

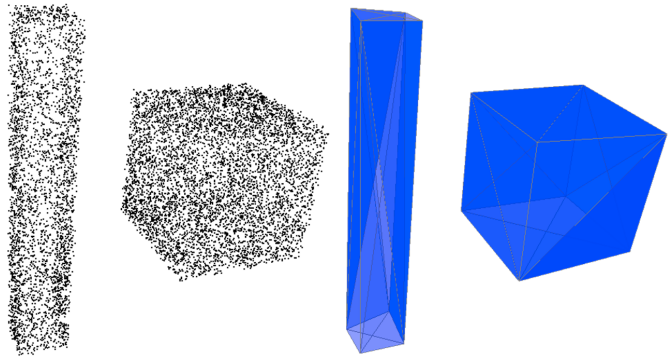


Figure 15: Reconstruction and simplification of a scene composed of two boxes. Left: $10K$ noisy sample points. Right: reconstruction with 16 vertices. The level of anisotropy matches our expectations but some facets of the boxes are covered twice.

described above. The feature recovery method is summed up in Algorithm 4.

Algorithm 4: Feature recovery.

Input : Point set $\mathcal{S}$, Reconstructed mesh $\mathcal{T}$.
Output: Modified mesh $\mathcal{T}$
Compute initial assignment;
Relocate all vertices;
Filter out facets of $\mathcal{T}$ by thresholding mass densities.

The input to the algorithm is a surface triangle mesh (the output of a smooth reconstruction algorithm) and the original point set used for reconstruction. Bins are initially sampled on the mesh. The initial assignment is performed through relaxation as described in Section 3.3: each sample is assigned to the nearest mesh vertex, and local reassignments are iterated until a local minimum for the transport cost is reached. Each mesh vertex is then relocated as described in Section 4.3. Through this method sharpness is recovered (Fig. 16).

For open surfaces this method allows for recovering the boundaries of the surface through the last filtering step (section 4.4) as can be seen on the church example (Figure 18 and 19). On the latter, the relocation seems incorrect at first glance on the bell tower, but after a more careful examination of the point set the spurious triangles created by the relocation procedure

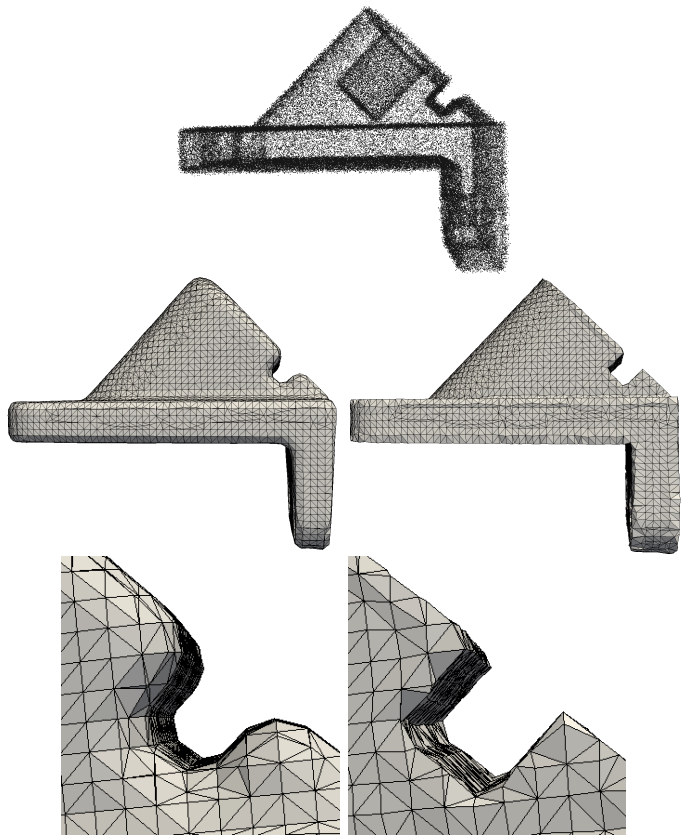


Figure 16: Anchor. Noisy point set (top), Poisson reconstruction (middle left), improved reconstruction (middle right) and associated closeups.

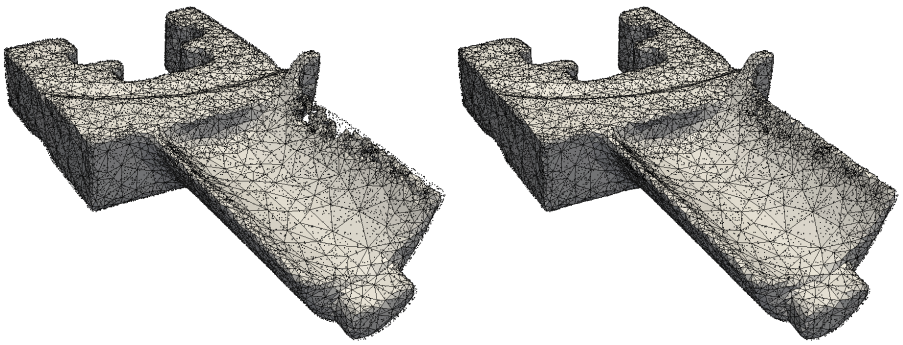


Figure 17: Blade. Poisson reconstruction (left) and improved reconstruction (right). The input point set is depicted with black dots. No remeshing nor edge flips are applied: the spurious topological handles shown in Figure 10 are not repaired as the triangles are only pulled toward the point set.

correspond to actual samples of the point set. These are details of the shape that were lost by the Poisson Reconstruction. On the challenging synthetic point set used in Figure 20 the vertex relocation tends to recover the sharpness of the features. In terms of computational cost applying the vertex relocation algorithm on the church mesh (23K vertices, 232K input points) takes around 10 minutes.

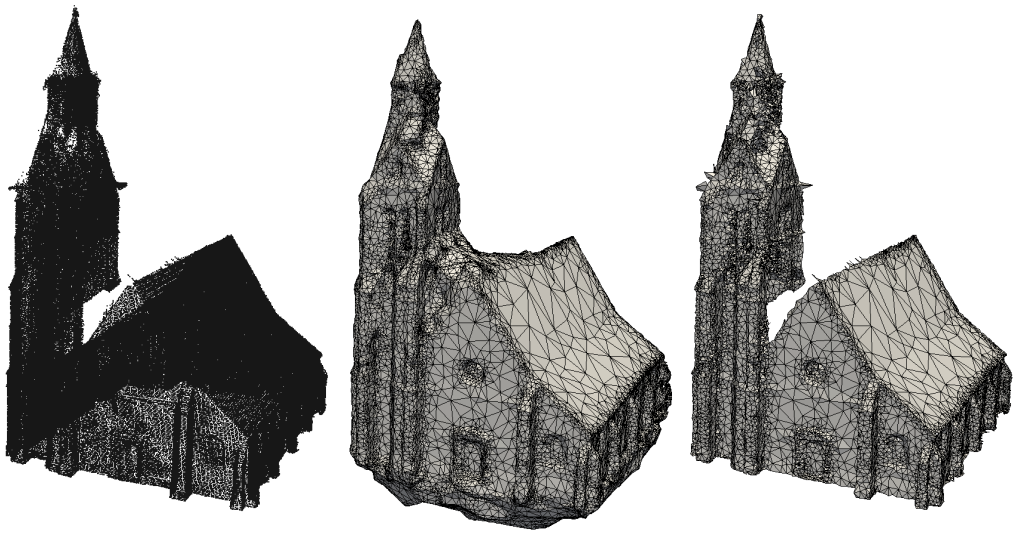


Figure 18: Church. Point set (left), Poisson reconstruction (middle) and relocated mesh (right).

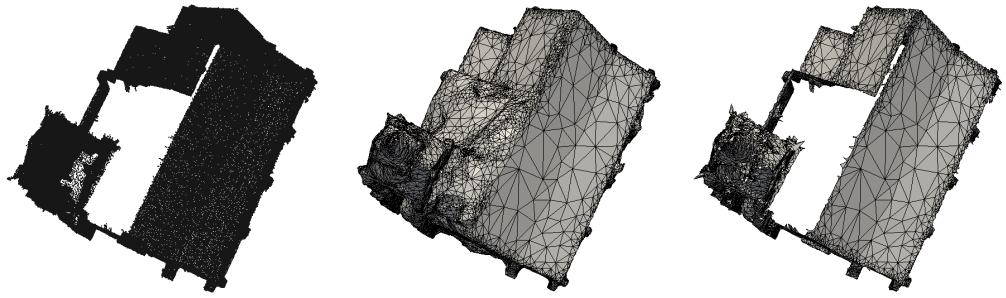


Figure 19: Church. Point set, Poisson reconstruction and vertex relocation. Filtering combined with vertex relocation allows recovering he surface boundaries.

6 Conclusion

We introduced a surface reconstruction and simplification method which exhibits both robustness to noise and outliers, as well as preservation of sharp features and boundaries. Our approach is based on the decimation of a simplicial complex guided by an optimal transportation error

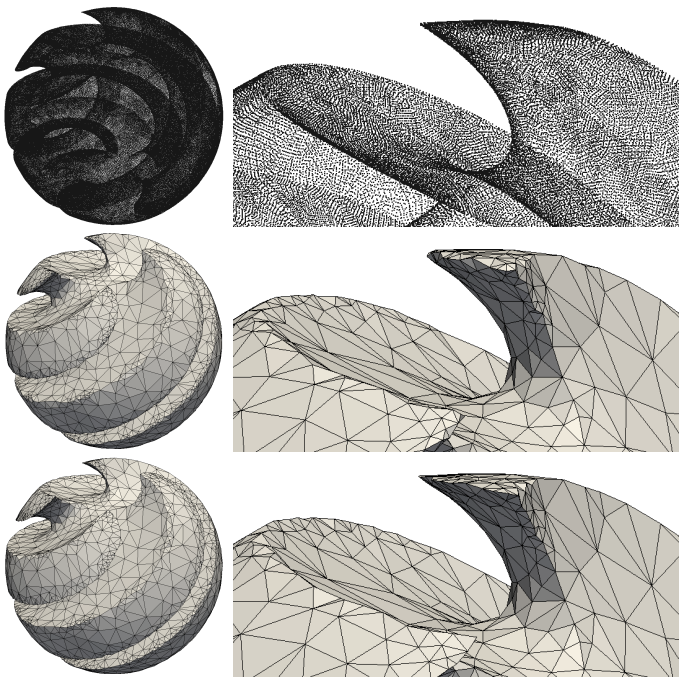


Figure 20: Sharp sphere. Left column: global view, right column: close-up. From top to bottom: Point set, smooth reconstruction and vertex relocation. Features are recovered through vertex relocation.

metric between the reconstruction and the initial point set. The metric proves also successful for feature recovery after a reconstruction method producing smooth shapes.

The main drawback of our approach is its computational cost: despite our efforts to introduce local relaxation, parallelization, and multiple-choice accelerations, we cannot reconstruct large point sets in reasonable time. The main strength of our approach lies into the simplicity of its formulation, which is expressed directly on the simplicial complex being reconstructed. This departs from common robust operators which require subsequent contouring to obtain the final reconstructed (but not simplified) surface mesh. In addition, our formulation allows the reconstruction of shapes of various dimensions as, *e.g.*, a curve embedded in $\mathbb{R}^3$ would be automatically reconstructed as a set of edges with a large majority of edge bins. Furthermore our formulation provides us with a transport plan, which can be used for further geometry processing of the resulting simplicial complex.

As future work we wish to improve scalability. Reusing the multi-scale approach of Mérigot [26] is certainly a valid direction but requires significant work to be fully practical.

References

[1] Adamson, A., Alexa, M.: Anisotropic point set surfaces. In: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, p. 13 (2006)

[2] Alliez, P., Cohen-Steiner, D., Tong, Y., Desbrun, M.: Voronoi-based variational reconstruction of unoriented point sets. In: EUROGRAPHICS Symposium on Geometry Processing, pp. 39–48 (2007)

[3] Amenta, N.: The crust algorithm for 3d surface reconstruction. In: Proceedings of the fifteenth annual symposium on Computational geometry, SCG '99, pp. 423–424. ACM (1999)

[4] Avron, H., Sharf, A., Greif, C., Cohen-Or, D.: ℓ_1-sparse reconstruction of sharp point set surfaces. ACM Trans. on Graphics **29**(5), 1–12 (2010)

[5] Bonneel, N., van de Panne, M., Paris, S., Heidrich, W.: Displacement interpolation using lagrangian mass transport. ACM Transactions on Graphics (SIGGRAPH Asia Proceedings) (2011)

[6] CGAL, Computational Geometry Algorithms Library. Http://www.cgal.org

[7] CLP, coin-or linear program solver. Http://www.coin-or.org/Clp/

[8] Daniels, J.I., Ha, L.K., Ochotta, T., Silva, C.T.: Robust smooth feature extraction from point clouds. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications, pp. 123–136 (2007)

[9] Digne, J., Morel, J.M., Souzani, C.M., Lartigue, C.: Scale space meshing of raw data point sets. Computer Graphics Forum **30**(6), 1630–1642 (2011)

[10] Dinh, H.Q., Turk, G., Slabaugh, G.: Reconstructing surfaces using anisotropic basis functions. In: Proceedings of the Eighth International Conference On Computer Vision, pp. 606–613 (2001)

[11] Du, Q., Faber, V., Gunzburger, M.: Centroidal voronoi tessellations: Applications and algorithms. SIAM Rev. **41**(4), 637–676 (1999)

[12] Fleishman, S., Cohen-Or, D., Silva, C.: Robust moving least-squares fitting with sharp features. In: ACM SIGGRAPH 2005 Papers, p. 552 (2005)

[13] Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, pp. 209–216 (1997)

[14] de Goes, F., Cohen-Steiner, D., Alliez, P., Desbrun, M.: An optimal transport approach to robust reconstruction and simplification of 2d shapes. Computer Graphics Forum **30**(5), 1593–1602 (2011)

[15] Gumhold, S., Wang, X., MacLeod, R.: Feature extraction from point clouds. In: Proceedings of 10th International Meshing Roundtable, pp. 293–305 (2001)

[16] Hornung, A., Kobbelt, L.: Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In: EUROGRAPHICS Symposium on Geometry Processing, pp. 41–50 (2006)

[17] Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. ACM Transactions on Graphics **28**(5) (2009)

[18] Jenke, P., Wand, M., Straßer, W.: Patch-graph reconstruction for piecewise smooth surfaces. Vision, modeling, and visualization 2008: proceedings p. 3 (2008)

[19] Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth EUROGRAPHICS Symposium on Geometry Processing, SGP '06, pp. 61–70 (2006)

[20] Kolluri, R., Shewchuk, J.R., O'Brien, J.F.: Spectral surface reconstruction from noisy point clouds. In: EUROGRAPHICS Symposium on Geometry Processing, pp. 11–21 (2004)

[21] Labatut, P., Pons, J.P., Keriven, R.: Robust and efficient surface reconstruction from range data. Computer Graphics Forum **28**(8), 2275–2290 (2009)

[22] Lindstrom, P., Turk, G.: Evaluation of memoryless simplification. IEEE Transactions on Visualization and Computer Graphics **5**(2), 98–115 (1999)

[23] Lipman, Y., Cohen-Or, D., Levin, D.: Data-dependent MLS for faithful surface approximation. In: Proceedings of the EUROGRAPHICS symposium on Geometry processing, p. 67 (2007)

[24] Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization free projection for geometry reconstruction. ACM Transactions on Graphics **26**(3), 22 (2007)

[25] Lipman, Y., Daubechies, I.: Surface comparison with mass transportation (2010). ArXiv preprint 0912.3488

[26] Mérigot, Q.: A multiscale approach to optimal transport. Computer Graphics Forum **30**(5), 1583–1592 (2011)

[27] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. In: Proc. of ACM SIGGRAPH, vol. 22(3), pp. 463–470 (2003)

[28] Oztireli, C., Guennebaud, G., Gross, M.: Feature preserving point set surfaces based on non-linear kernel regression. In: Computer Graphics Forum, vol. 28(2), pp. 493–501 (2009)

[29] Pang, X.F., Pang, M.Y.: An algorithm for extracting geometric features from point cloud. International Conference on Information Management, Innovation Management and Industrial Engineering **4**, 78–83 (2009)

[30] Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled surfaces. Computer Graphics Forum **22**(3), 281–289 (2003)

[31] Peyré, G., Fadili, J., Rabin, J.: Wasserstein active contours. Tech. rep., Preprint Hal-00593424 (2011). URL `http://hal.archives-ouvertes.fr/hal-00593424/`

[32] Rabin, J., Delon, J., Gousseau, Y.: Regularization of transportation maps for color and contrast transfer. In: Image Processing (ICIP), 2010 17th IEEE International Conference on, pp. 1933 –1936 (2010)

[33] Rabin, J., Delon, J., Gousseau, Y.: Transportation distances on the circle. J. Math. Imaging Vis. **41**(1-2), 147–167 (2011)

[34] Rabin, J., Peyré, G., Cohen, L.D.: Geodesic shape retrieval via optimal mass transport. In: Proceedings of the 11th European conference on Computer vision: Part V, ECCV'10, pp. 771–784. Springer-Verlag, Berlin, Heidelberg (2010)

[35] Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: A. Bruckstein, B. ter Haar Romeny, A. Bronstein, M. Bronstein (eds.) Scale Space and Variational Methods in Computer Vision, *Lecture Notes in Computer Science*, vol. 6667, pp. 435–446. Springer Berlin / Heidelberg (2012)

[36] Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. **21**(5), 34–41 (2001)

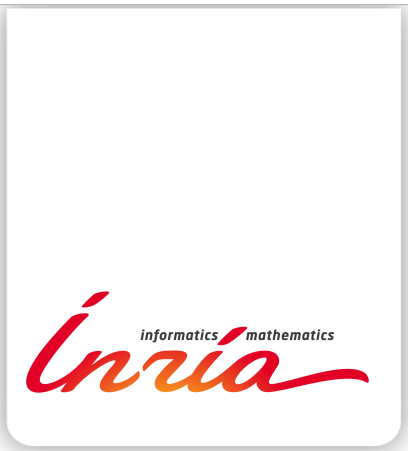
[37] Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vision **40**(2), 99–121 (2000)

[38] Salman, N., Yvinec, M., Mérigot, Q.: Feature Preserving Mesh Generation from 3D Point Clouds. In: Computer Graphics Forum, vol. 29, pp. 1623–1632 (2010)

[39] Villani, C.: Topics in Optimal Transportation. American Mathematical Society (2010)

[40] Walder, C., Chapelle, O., Schölkopf, B.: Implicit surface modelling as an eigenvalue problem. In: Machine Learning ICML 2005, pp. 936–939 (2005)

[41] Wu, J., Kobbelt, L.: Fast mesh decimation by multiple-choice techniques. In: Proceedings of the Vision, Modeling, and Visualization Conference, pp. 241–248 (2002)



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399